A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering

Supplementary Material
Paper ID 3167

1. Question generation stoplist

When selecting words to blank to create fill-in-the-blank question-answer pairs, we performed some manual filtering to prevent undesirable words from becoming targets. This is the list of words which were disallowed for blanking (i.e., the stoplist):

A	Dr	Elsewhere
IN	Meanwhile	Mr
Mrs	Now	OF
SOMEONE	SOMEONE’	SOMEONE’s
THE	a	as
aside	be	before
can’t	does	doesn’t
in	is	it
it’s	it’s	later
meanwhile	new	not
now	on	other
other’s	something	them
there	up	while
who	who’s	

2. Model Specification

We describe the different hyperparameters used in our experimental setting.

2.1. Question Encoding

Words composing a question are given as in one-hot encoding formats, leading to vectors having the same size than the vocabulary. They are then projected into vector of size 512 using an embedding matrix. They are fed to the forward and backward BN-LSTM leading to hidden vectors of size 320. The hidden vectors corresponding to the last timestep of the forward and backward BN-LSTM are concatenanted to obtain the question respresentation.

2.2. Video Encoding

As specified in the main paper, we rely on a GoogLeNet convolutional neural network that has been pretrained on ImageNet [3] to extract static features. Features are extracted from the pool5/7x7 layer. 3D moving features are extracted using the C3D model [4], pretrained on Sport 1 million [1]. We apply the C3D frames on chunk of 16 consecutive frames in a video and retrieve the activations corresponding to the “fc7” layer. We don’t finetune the 2D and 3D CNN parameters during training on the fill-in-the-blank task. It leads to an input visual vector of size 1024 for GoogleNet, 4096 for C3D hence 5120 when we are concatenating both GoogleNet and C3D. If not specified otherwise, we extract visual vectors for 25 temporal chunks in the videos.

Vectors are then fed to a video encoding BN-LSTM of size 320. The hidden vectors corresponding to the last timestep of the video LSTM is used as video representation.

2.3. Classifier

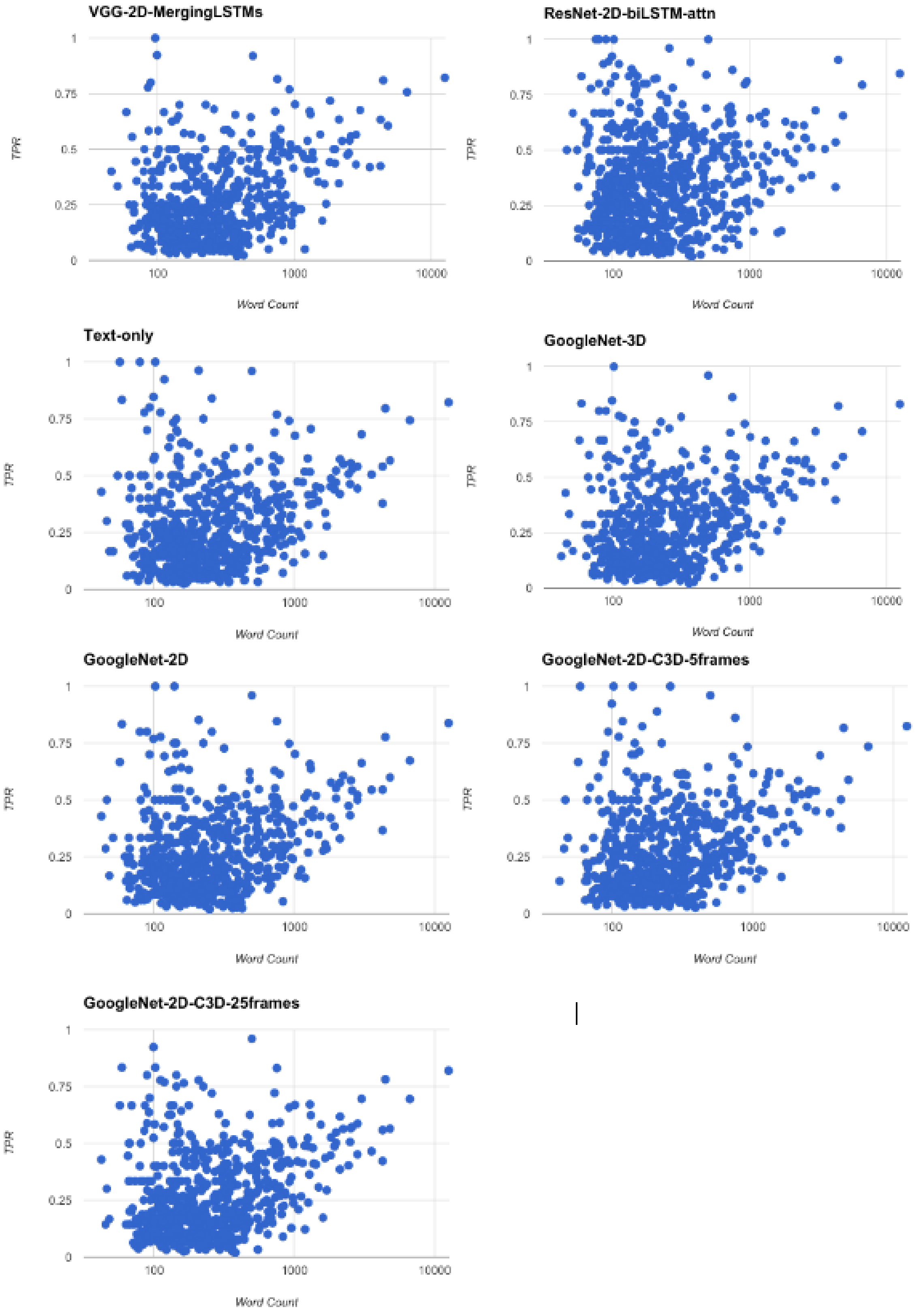
The classifier is a single layer MLP with a softmax activation function in order to output a probability distributions over the diverent candidate words. We apply dropout on the MLP inputs with drop probability of 0.5.

2.4. Training

We used Adam [2] with the gradient computed by the backpropagation algorithm. We use a learning rate of $1e{-}3$, and a gradient clipping of 10. Early stopping of the model is performed based on the validation set performance.

3. True Positive Rate (TPR) by Word Frequency for all models

In the main paper we plot TPR (number of times a model correctly filled in the blank with a given word) by word frequency (how often that word appears in the training set) for GoogleNet-2D-C3D; here in Figure 1 we provide these plots for all models. These plots show that all models perform better on more frequent words, suggesting that increasing dataset size would be one way to improve model performance.


VGG-2D-MergingLSTMs

ResNet-2D-biLSTM-attn

Text-only

GoogleNet-3D

GoogleNet-2D

GoogleNet-2D-C3D-5frames

GoogleNet-2D-C3D-25frames

Figure 1. True Positive Rate (TPR) per Word Frequency for all models.

References

[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[2] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[4] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.